Počítačové zpracování přirozeného jazyka

Morfologická analýza Unifikační gramatiky

Daniel Zeman

http://ufal.mff.cuni.cz/course/popj1/

Unifikační gramatiky

- Založeny na
  - bezkontextových gramatikách
  - **strukturách rysů** (feature structures)
  - jejich **unifikovatelnosti**
- Struktura rysů
  - Něco jako záznam v databázi, naplněná proměnná typu **`record`** v pascalu nebo typu **`struct`** v céčku. Popis objektu, seznam rysů.
  - rysy (atributy, features) … jména polí, položek
  - hodnoty
  - Příklady dvojic rys – hodnota: [číslo: množné], [pád: 1].

Struktura rysů

entita	
JMÉNO	FF UK
TELEFON	258562

POS	substantivum
GEN	maskulinum
NUM	singulár
CASE	dativ

entita	
JMÉNO	Dan
TELEFON	221914225

POS	adjektivum
GEN	maskulinum
NUM	plurál
CASE	akuzativ
DEG	komparativ
NEG	afirmativ

fakulta	
JMÉNO	MFF UK
DĚKAN	Netuka
TELEFON	221911111

Struktura rysů

- Obecně: parciální funkce z množiny rysů do množiny hodnot.

typ	
$\mathtt{RYS}_1$	$\mathtt{HODNOTA}_1$
$\mathtt{RYS}_2$	$\mathtt{HODNOTA}_2$
$\mathtt{RYS}_3$	$\mathtt{HODNOTA}_3$

Unifikovatelnost

- Dvě struktury rysů jsou **unifikovatelné**, jestliže se shodují jejich hodnoty v těch rysech, které mají obě dvě.
- Příklad: struktury 1 a 2 jsou unifikovatelné, 2 a 3 také, 1 a 3 nikoliv.

1

ROD	mužský
ČÍSLO	jednotné
PÁD	dativ

2

DRUH	sloveso
ČÍSLO	jednotné
ČAS	přítomný

3

ROD	mužský
ČÍSLO	jednotné
PÁD	instrumentál

Unifikace

- **Unifikace** je operace na dvou unifikovatelných strukturách rysů, jejím výsledkem je nová struktura rysů.

$$1\begin{bmatrix}\texttt{ROD} & \texttt{mužský}\\ \texttt{ČÍSLO} & \texttt{jednotné}\\ \texttt{PÁD} & \texttt{dativ}\end{bmatrix} + 2\begin{bmatrix}\texttt{OSOBA} & \texttt{třetí}\\ \texttt{ČÍSLO} & \texttt{jednotné}\\ \texttt{ČAS} & \texttt{přítomný}\end{bmatrix}$$

$$= 3\begin{bmatrix}\texttt{ROD} & \texttt{mužský}\\ \texttt{ČÍSLO} & \texttt{jednotné}\\ \texttt{PÁD} & \texttt{dativ}\\ \texttt{OSOBA} & \texttt{třetí}\\ \texttt{ČAS} & \texttt{přítomný}\end{bmatrix}$$

Morfologická **syntéza** pomocí unifikace

- Vstup: struktury rysů „lemma“ a „značka“.
- Najít ve slovníku všechny struktury „heslo“, které jsou unifikovatelné se strukturou „lemma“.
- Pro každou nalezenou strukturu „heslo“ najít v seznamu vzorů strukturu „vzor“, která je současně unifikovatelná s ní i se vstupní strukturou „značka“.
- Unifikací k sobě pasujících struktur „heslo“, „vzor“ a „značka“ vznikne struktura „tvar“.
- Na výstupu je pro každou strukturu „tvar“ řetězec složený z hodnot jejích rysů „vzor“ a „koncovka“.

Morfologická syntéza pomocí unifikace

- Vstup: struktury rysů „lemma“ a „značka“.

```
lemma
LEMMA        háček
```

```
značka
ČÍSLO        množné
PÁD          nominativ
```

- Najít ve slovníku všechny struktury „heslo“, které jsou unifikovatelné se strukturou „lemma“.

```
heslo
LEMMA        háček
VZOR         hrad
```

```
heslo
LEMMA        háček
VZOR         pán
```

Morfologická syntéza pomocí unifikace

- Pro každou nalezenou strukturu „heslo“ najít v seznamu vzorů strukturu „vzor“, která je současně unifikovatelná s ní i se vstupní strukturou „značka“.

```
heslo
LEMMA      háček
VZOR       hrad
```

```
heslo
LEMMA      háček
VZOR       pán
```

```
vzor
VZOR       hrad
ČÍSLO      množné
PÁD        nominativ
KONCOVKA   y
```

```
vzor
VZOR       pán
ČÍSLO      množné
PÁD        nominativ
KONCOVKA   i | ové
```

Morfologická syntéza pomocí unifikace

- Unifikací k sobě pasujících struktur „heslo“, „vzor“ a „značka“ vznikne struktura „tvar“.

```
tvar
LEMMA      háček
VZOR       hrad | pán
ČÍSLO      množné
PÁD        nominativ
KONCOVKA   y | i | ové
```

Morfologická syntéza pomocí unifikace: poznámky

- Unifikace se podobá databázovým operacím.
- Sama neříká, jak ze struktury „tvar“ vznikne slovní tvar.
- Pravidlo:
  `výstup = tvar.lemma + tvar.koncovka`
- Zesložitění pravidla, aby řešilo fonologicky podmíněné změny (na to už je unifikace nevhodná):

místo

**háčeky, *háčeki, *háčekové*

chceme

háčky, háčci, háčkové

Morfologická **analýza** pomocí unifikace

- Neunifikační část: najít všechny možné afixy, které lze ve slově vidět ⇒ množina struktur „tvar“.
- Které afixy (koncovky) existují, víme ze struktur typu „vzor“.
- Vyřešit (nějak) změny kmenových souhlásek, palatalizaci apod.
- Pak už lze postupovat opačně k syntéze: unifikovat tvar se vzorem, a výsledek se slovníkem. Co se ve slovníku skutečně najde, patří do analýzy.
  - např. běžím=běžet(trpět)+osoba(1), ≠běží(stavení)+pád(7)

Unifikační morfologická gramatika (UMG)

- Jan Hajič: *Unification Morphology Grammar (doktorandská práce)*. Univerzita Karlova, Praha, 1994
- Stuart Shieber: *An Introduction to Unification-based Approaches to Grammar*. CSLI Lecture Notes No. 4, Stanford, California, USA, 1986

- Základ: bezkontextová gramatika.
- S každou složkou (symbol + rozsah) je spojena struktura rysů.
- Pravidlo: levá strana → pravá strana := operace nad strukturami rysů.
- Operace mohou i zablokovat použití pravidla požadavkem na unifikovatelnost.
- Unification-based chart parser, PATR-II (Shieber).
- Podobně jako CFG byly i unifikační gramatiky původně navrženy pro analýzu věty a teprve později aplikovány na analýzu slova.

Syntax UMG

- Levá → pravá **:=** operace se strukturami rysů
  - pravidlo gramatiky
- **<X>**
  - neterminál X. Terminály se píší jen tak.
- **#**
  - operátor unifikace (současně vznáší požadavek na unifikovatelnost)
- **^**
  - operátor odkazu (odděluje části cesty (neterminály) ke struktuře rysů, na kterou odkazujeme)
- **+**
  - operátor sřetězení
- **|**
  - operátor disjunkce. Z disjunkce struktur rysů se vyberou všechny struktury, které lze použít (jsou unifikovatelné atd.) Disjunkce struktur rysů může zachytit různé analýzy téhož řetězce.

Příklad pravidla v UMG

<N> → <L> := [l = <L>^l, umlaut = <L>^umlaut # no]

- Interpretace:
  - Pokud:
    - rozpoznali jsme složku **<L>** a
    - hodnota atributu **umlaut** ve struktuře rysů svázané s touto složkou je „no“
  - Potom:
    - rozpoznali jsme ve stejném rozsahu složku **<N>**
    - do její struktury rysů máme ze struktury rysů složky **<L>** okopírovat atributy **l** a **umlaut**

Teoretický pohled na slovník

- Pravidlo, které generuje prázdný řetězec, ale poskytuje své levé straně obrovskou strukturu rysů, obsahující celý slovník.
  - **<LEX> → "" :=**
    [kmen=mat, hw=matka, pos=N, x=zn6e] |
    [kmen=atom, hw=atom, pos=N, x=hd1] |
    [kmen=nov, hw=nový, pos=A, x=reg] |
    [kmen=prac, hw=pracovat, pos=V,
    x=ovatn] |
    …;

Teoretický pohled na slovník

- Napojení slovníku na zbytek gramatiky:
  - **<R> → <S>u <LEX> := <LEX> # [x=hd1, kmen=<S>, case=gen|dat|loc, num=sg]**
    - Pravidlo reprezentuje tvoření 2., 3. a 6. pádu jednotného čísla podle vzoru **hd1** (hrad).
    - **R** zastupuje slovo unifikované se slovníkem.
    - **S** je část vstupu odpovídající kmeni slova. Koncovka je uvedena přímo terminálem, **LEX** za ní odpovídá prázdnému řetězci.
    - Pravidlo za **:=** říká, že nás z **LEX** zajímají ty struktury, jejichž kmen odpovídá **S** a kódují 1. nebo 4. pád jednotného čísla podle vzoru hd1.
    - Slovníkové záznamy, které projdou tímto filtrem, utvoří množinu struktur rysů svázanou s neterminálem **R**. Navíc se do těchto struktur připíše informace o čísle a pádu.

Příklad UMG

```
<L> → a := [l=a];
<L> → b := [l=b];
…
<N> → <L> := [l=<L>^l];
<N> → <L> <N> := [l=<L>^l+<N>^l];
<S> → <N> := <N>;
<R> → <S> := <LEX> # [stem=<S>^l, x=hd1, num=sg,
  case=nom|acc, …];
<R> → <S>u := <LEX> # [stem=<S>^l, x=hd1, num=sg,
  case=gen, …];
<LEX> → "" := … | [stem=hrad, x=hd1, …] | …
```

Příl[illegible]G

<L> je písmeno

<N> je řetězec

<S> je potenciální kmen slova

<R> je rozpoznaný tvar slova ověřený ve slovníku

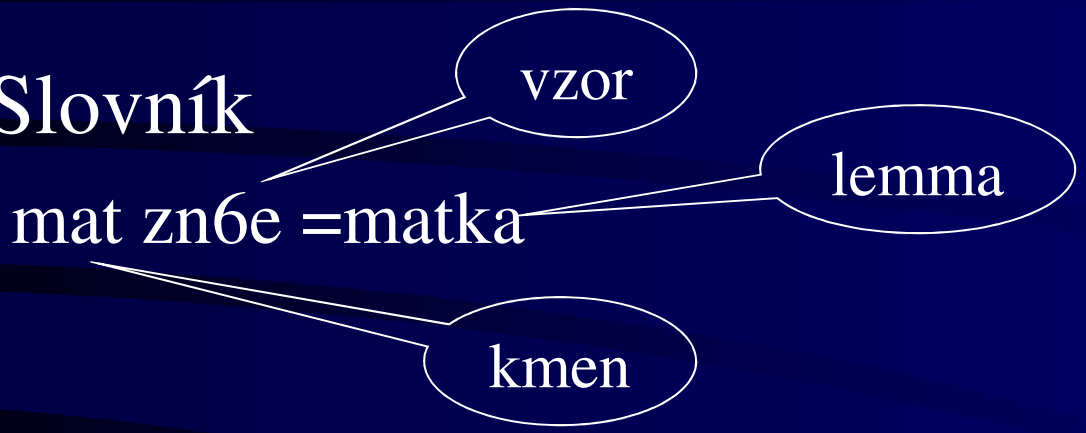
```
<L> → a := [l=a];
<L> → b := [l=b
…
<N> → <L> := [l=<L>^l];
<N> → <L> <N> := [l=<L>^l+<N>^l];
<S> → <N> := <N>;
<R> → <S> := <LEX> # [stem=<S>^l, x=hd1, num=sg,
   case=nom|acc, …];
<R> → <S>u := <LEX> # [stem=<S>^l, x=hd1, num=sg,
   case=gen, …];
<LEX> → "" := … | [s…
```

Slovník v praxi

- Začlenění do gramatiky není efektivní.
- V praxi se obchází:
  - Slovník uložit v samostatné datové struktuře s efektivním vyhledáváním.
  - Pravidla obsahující **<LEX>** ošetřit algoritmem pro přístup k této struktuře.
  - Zbytek gramatiky zpracovat normální analýzou.

Příklad UMG

- Slovník



Typický systém s mnoha vzory, např. 44 různých vzorů odpovídá „školnímu“ vzoru *žena* (aniž by vzory řešily případné zkracování kmenové samohlásky).

Příklad UMG

{ vzor = stavení; levá strana je vždy stejná, vynecháváme }

```
<_><í>$ := [key=<_>í, x=(st|rž), cat=[pos=n],
  morf=[infl=[pf=([gnd=n, num=sg,
  case=(nom|gen|dat|acc|voc|loc)] | [gnd=n,
  num=pl, case=(nom|gen|acc|voc)])]]];
<_><í><m>$ := [key=<_>í, x=(st|rž), cat=[pos=n],
  morf=[infl=[pf=([gnd=n, num=sg, case=ins] |
  [gnd=n, num=pl, case=dat])]]];
<_><í><c><h>$ := [key=<_>í, x=(st|rž),
  cat=[pos=n], morf=[infl=[pf=[gnd=n, num=pl,
  case=loc]]]];
<_><í><m><i>$ := [key=<_>í, x=(st|rž),
  cat=[pos=n], morf=[infl=[pf=[gnd=n, num=pl,
  case=ins]]]];
```

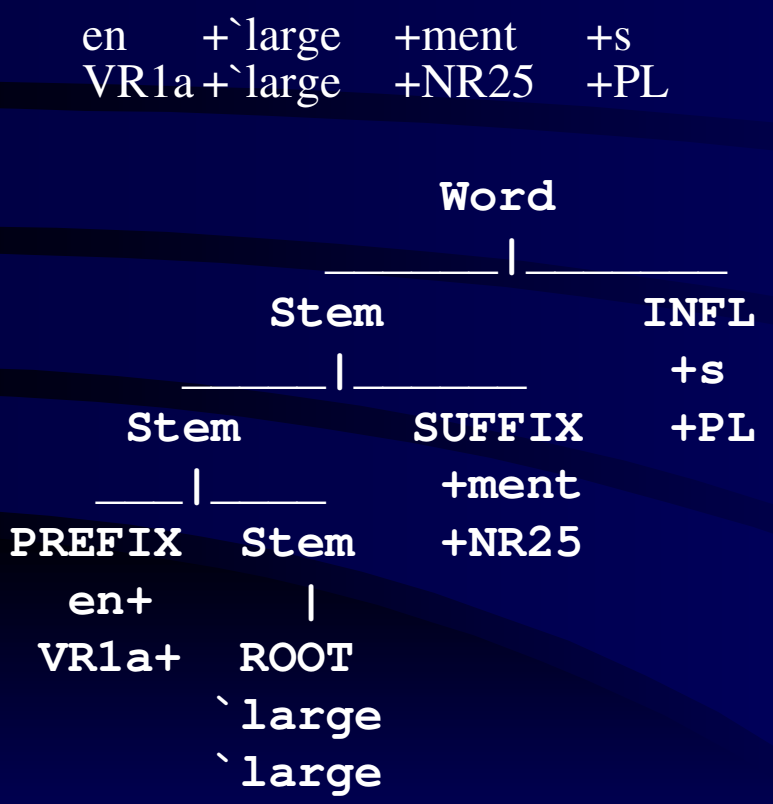
Srovnání UMG a CFG

- Struktura rysů uchovává výstup analýzy (značku) $\Rightarrow$ nepotřebujeme dohodu o pojmenování neterminálů
- Disjunkce struktur zachytí homonymní analýzy $\Rightarrow$ nepotřebujeme štěpit neterminály
- Fonologie stále problematická. Buď exploze vzorů (UMG), nebo kombinace s dvojúrovňovými pravidly (viz dále)

PC-Kimmo Word Grammar

- Unifikační gramatika podle Stuarta Shiebera. Trochu jiná syntax než UMG, podobné použití.
- lexicon
  - rozpoznání morfémů ve slově
- rules
  - fonologické změny na hranici morfémů
- grammar
  - rozbor vztahů mezi morfémy
  - odvození vlastností slova z vlastností morfémů
  - omezující podmínky na to, které morfémy lze kombinovat

PC-Kimmo Word Grammar




```
Word:
[ cat:    Word
  head:     [ agr:
                  [ 3sg: - ]
              number:PL
              pos:    N ]
  root:   `large
  root_pos:AJ
  clitic:-
  drvstem:- ]
```

PC-Kimmo Word Grammar

- Stará část PC-Kimma nejdříve tokenizuje slovo na morfémy.
- Nová část potom rozebere posloupnost morfémů podle gramatiky.
  - Gramatika může některé posloupnosti morfémů zavrhnout.
  - Ostatním přiřadí výklad (strukturu rysů). Staré PC-Kimmo dokázalo glosovat morfémy, ale nedokázalo říct, co z toho plyne pro celek (např. že přípona *–able* udělá ze slovesa přídavné jméno).
- Takhle vypadá pravidlo gramatiky:
  - Word -> Stem INFL
    <Stem head pos> = <INFL from_pos>
    <Word head> = <INFL head>

Pravidlo gramatiky

- **`Word -> Stem INFL`**
  `<Stem head pos> = <INFL from_pos>`
  `<Word head> = <INFL head>`

- Pravidlo nelze použít, jestliže rys pos podstruktury head morfému Stem není roven rysu from_pos morfému INFL.
  - Symboly morfémů jsou preterminály a odpovídají názvům podslovníků, ve kterých byly morfémy nalezeny.

- Pokud bude pravidlo použito, má se hodnota rysu head ze složky INFL zkopírovat do stejnojmenného rysu složky Word.

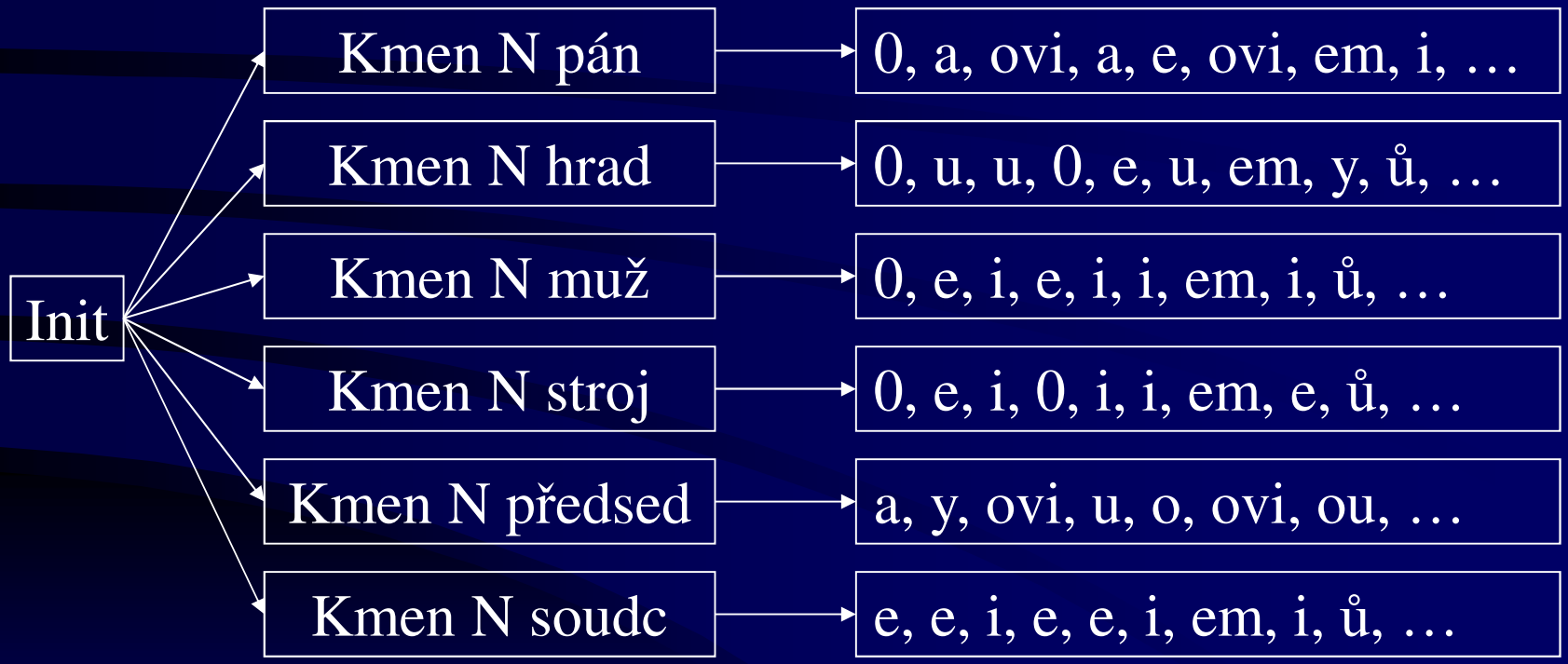
Pravidlo gramatiky

- **RULE** <pravidlo>
  <podmínky pravidla>
- Levou a pravou stranu pravidla odděluje **->** nebo **=**.
- **RULE Stem_1 = Stem_2 SUFFIX**
- **X** zastupuje libovolný terminál nebo neterminál.
- Zvláštní znaky jsou **()[]{}<>=:/**
  - Podtržítko jen pro připojení indexu k symbolu.
- Levá strana prvního pravidla je počáteční symbol gramatiky.
- **N = Nstem {Sing / Plural}**

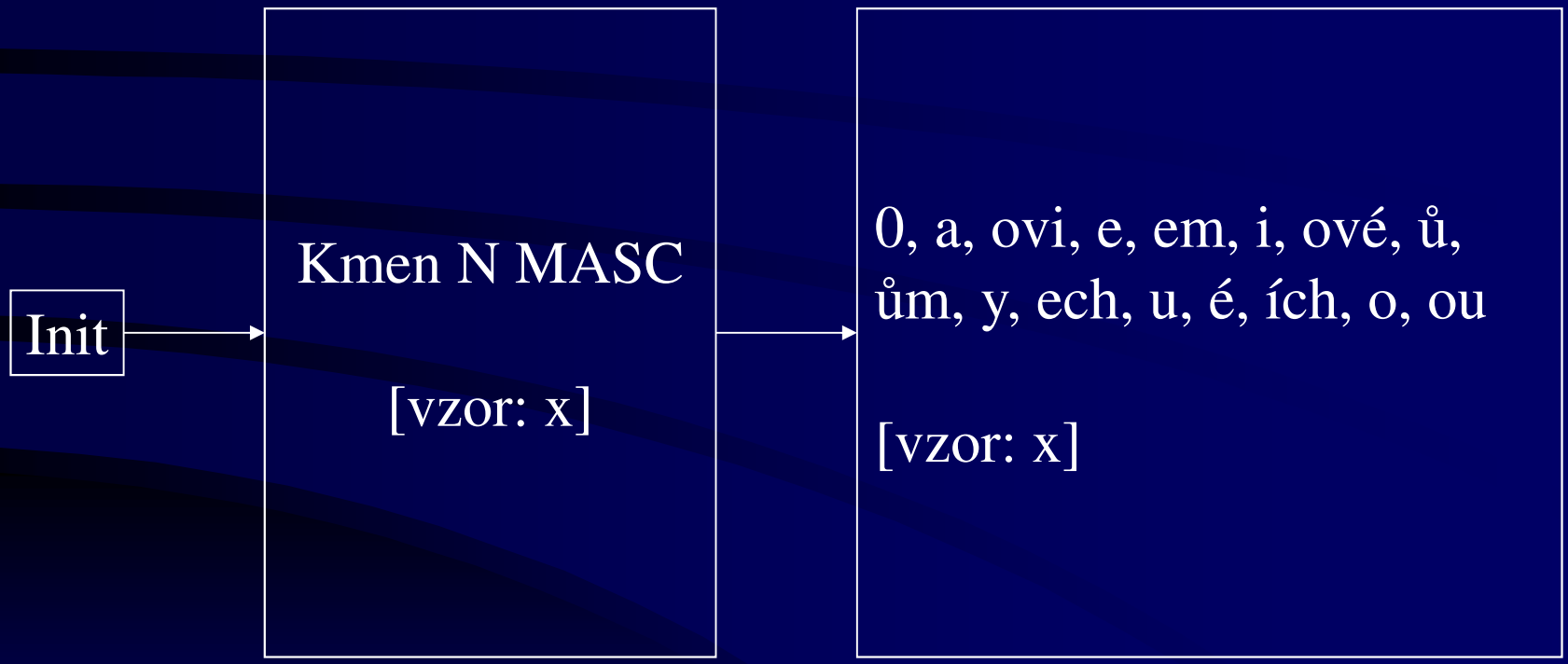
Výhody gramatiky

- Příklady z češtiny:
  - Gramatika zabrání spojení kmene podle vzoru „žena“ s koncovkou podle vzoru „růže“.
  - Ohlídá i závislosti na dlouhé vzdálenosti, např.:
    - nejchytřejší
  - Utvořením přídavného jména přivlastňovacího „ženin“ se změní rod z ženského na mužský. Původní rod se uloží jako rod vlastníka.

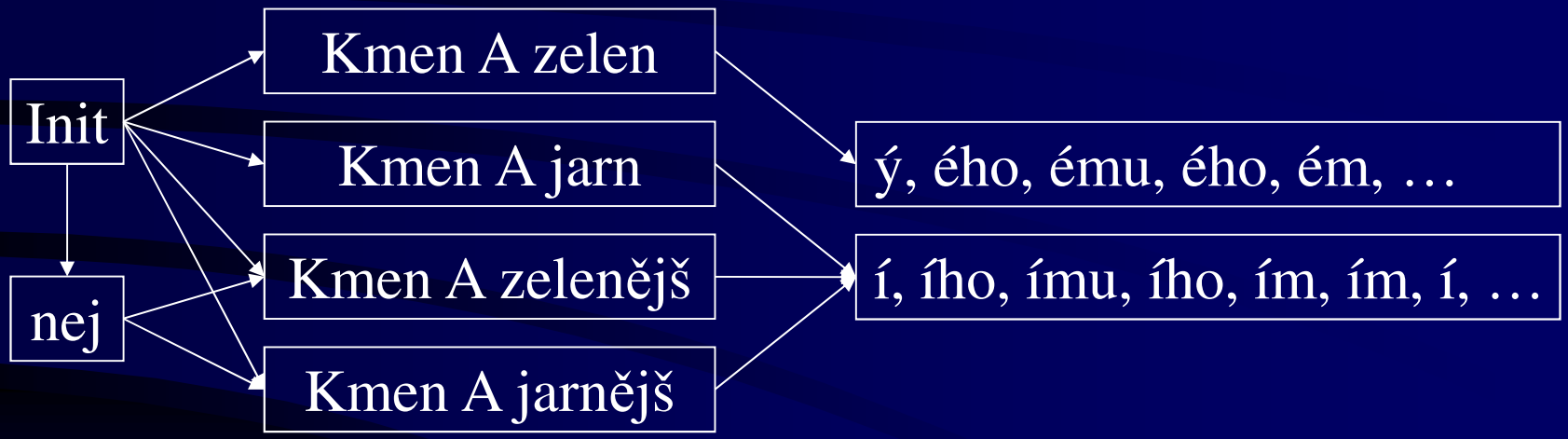
Bez gramatiky



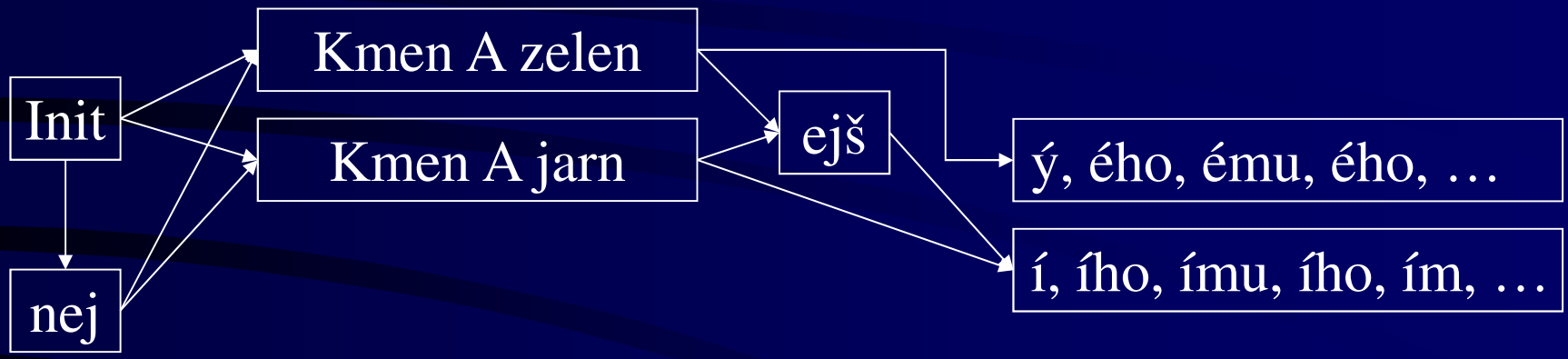
S gramatikou



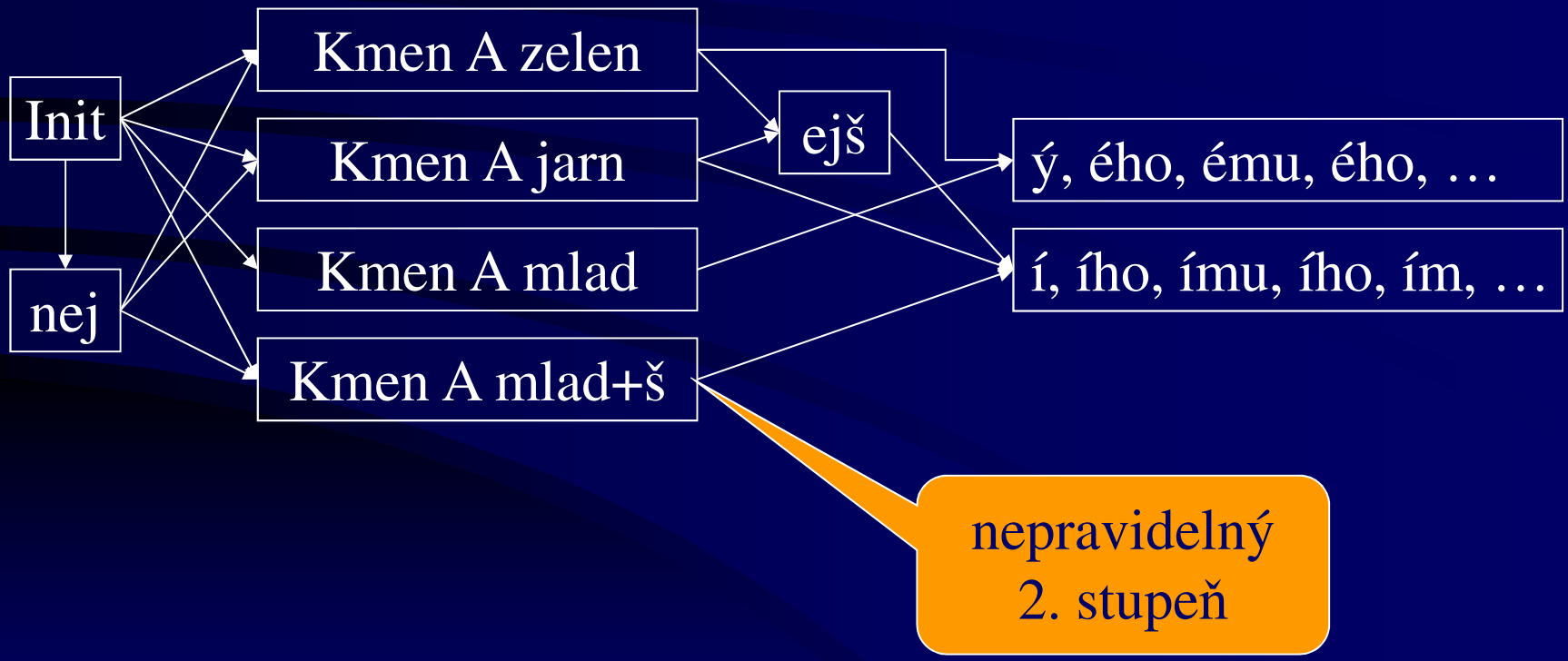
Bez gramatiky



S gramatikou



S gramatikou



Gramatika neovlivňuje fonologii

- Fonologické pravidlo pro změkčování v rozkazovacím způsobu:
  - meteš → meť(me,te)
  - t:ť ⇔ _ +:0 λ:0 *nebo* m:m e:e *nebo* t:t e:e
- Nemá se uplatnit ve 2. pádě množného čísla ženského rodu podstatných jmen:
  - kóta → *kóť
- Fonologická pravidla si nemohou ze struktur rysů přečíst, kdy se mají uplatnit.

Automatické rysy

- Každé slovníkové heslo má automaticky tyto základní rysy:
  - cat = název podslovníku (\lx)
  - lex = morfém, lexikální řetězec (\lf)
  - gloss = glosa ze slovníku (\gl)

Přiřazování hodnot rysům

- Zkratky přiřazení rysů
  - Hodnoty, které budeme přiřazovat každému slovníkovému heslu, chceme psát co nejkratší.
  - **LET** <zkratka | kategorie> **be** <definice>
  - např.
  - **Let pl be [number: PL]**
  - **Let pl be <number> = PL**
  - **Let 3sg be [tense: PRES**
    agr: 3SG]
- Disjunkce:
  - **Let sg/pl be {[number:SG][number:PL]}**
  - **Let sg/pl be <number> = {SG PL}**
- Výchozí hodnoty:
  - **Let N be <number> = !SG**
  - Nepřiřadí-li někdo podstatnému jménu explicitně číslo, má se za to, že číslo je jednotné.

Lexikální pravidla

- Nikoli zkratky, ale systematické transformace rysů pro skupiny slovníkových hesel. Převedou jednu strukturu rysů na jinou.
- **`DEFINE`** <název lexikálního pravidla> **`as`** <zobrazení>
- Příklad v dokumentaci na webu je vadný.
- Na konci analýzy, když už máme hotovou strukturu rysů pro celé slovo, můžeme aplikovat lexikální pravidlo, které tuto strukturu upraví.

Nastavení parametrů

- **PARAMETER** <název> **is** <hodnota>
  - **Parameter Start symbol is Word**
  - **Parameter Attribute order is cat head root**
    - V jakém pořadí má PC-Kimmo zobrazovat rysy?
  - **Category feature** (výchozí: **cat**)
  - **Lexical feature** (výchozí: **lex**)
  - **Gloss feature** (výchozí: **gloss**)
    - Jak se jmenují důležité rysy se zvláštním významem?

Ukázka v PC Kimmo

- `r ženě`
- Syntéza (nová v PCK v. 2, ale použití gramatiky není povinné)
- `l synthesis-lexicon cs.lex`
- `s N(žena) +SG+LOC`
- Je-li k dispozici gramatika, zablokuje syntézu nedovolených kombinací
- Nejde ale generovat ze struktury rysů